

# C\_C++程序设计编程环境搭建简单手册

操作系统：windows

V1.0

苏州大学计算机科学与技术学院

苏州大学软件学院

## 一、编程环境说明

1、编译器：

- a) x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0.7z：编译工具链都包含在这个压缩包里。这是 windows 环境下的基于 MinGW-w64 的 GNU 工具链。
  - b) MinGW-w64-install.exe 是基于 MinGW-w64 的 GNU 工具链在线安装程序，但由于网络的原因，在线安装很难成功。
- 后面编程环境搭建以 a) 方法介绍。

## 二、编程环境搭建

1、将 x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0.7z 解压到 x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0。如下图所示。



这里解压到完成目录下是便于显示编程环境的版本。也可以直接解压到本地目录，得到 mingw64 文件夹。

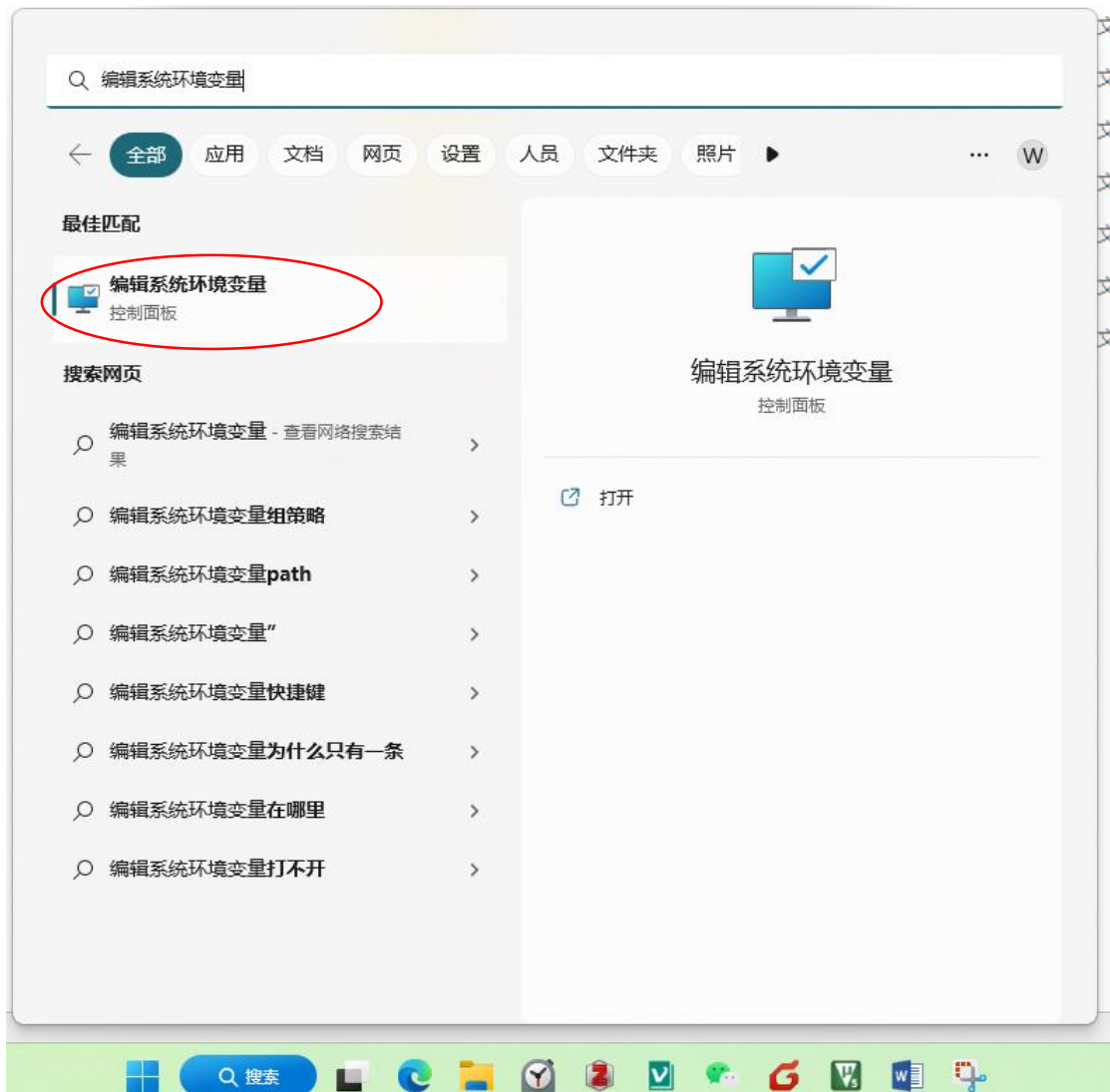
- 2、在硬盘上选择一个位置创建目录 mingw-w64。本文选择在 C 分区的"Program Files"目录下创建目录：C:\Program Files\mingw-w64。后面操作都是基于该目录路径。
- 3、将 x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0 整个文件夹拷贝到 C:\Program

Files\mingw-w64，如下图所示。

此电脑 > Windows-SSD (C:) > Program Files > mingw-w64 > x86_64-8.1.0-release-posix-seh-rt_v6-rev0 > mingw64				
	名称 ^	修改日期	类型	大小
Personal	bin	2023/2/5 16:09	文件夹	
	etc	2023/2/5 16:09	文件夹	
	include	2023/2/5 16:09	文件夹	
	lib	2023/2/5 16:09	文件夹	
	libexec	2023/2/5 16:09	文件夹	
	licenses	2023/2/5 16:09	文件夹	
	opt	2023/2/5 16:09	文件夹	
	share	2023/2/5 16:09	文件夹	
	x86_64-w64-mingw32	2023/2/5 16:09	文件夹	
	build-info.txt	2018/5/12 15:29	文本文档	49 KB

#### 4、设置环境变量。(以 win11 为例)

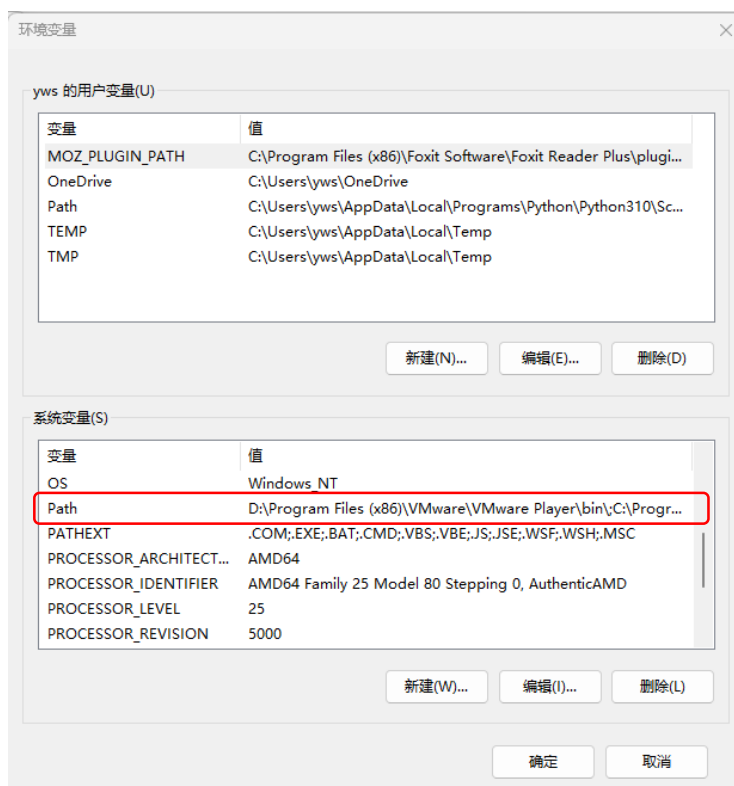
在开始菜单里输入“编辑系统环境变量”，如下图所示。



5、点击编辑系统环境变量应用激活窗口，如下图所示。

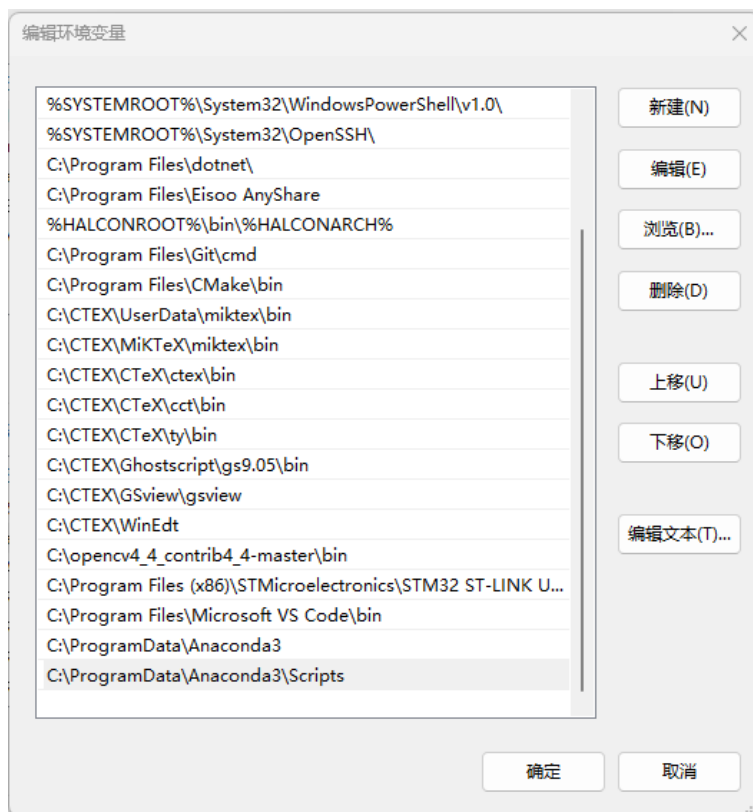


6、点击环境变量按钮打开环境变量查看对话框，如下图所示。

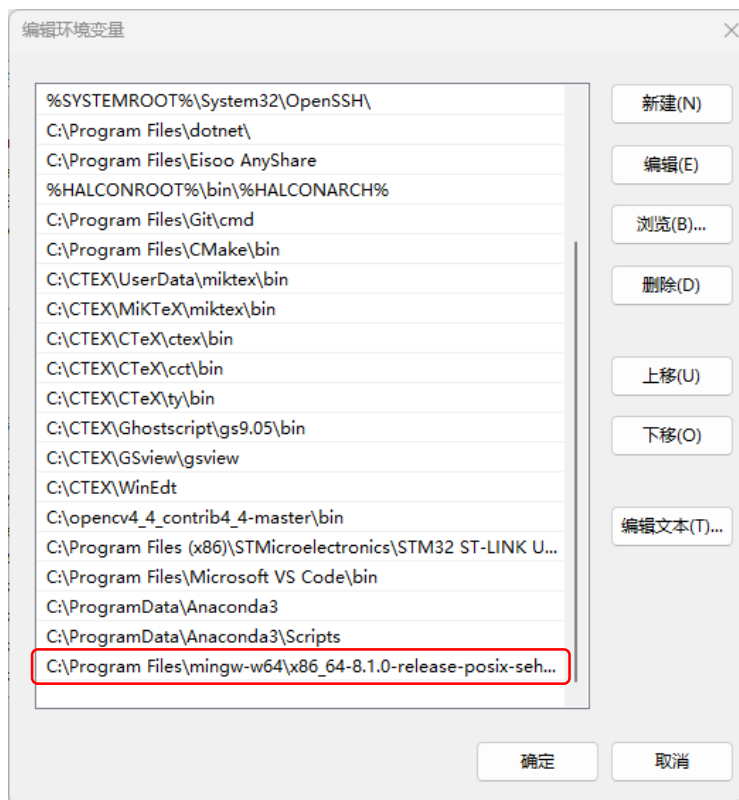


这里有用户变量和系统变量。在系统变量里找到 Path 变量，并选择。

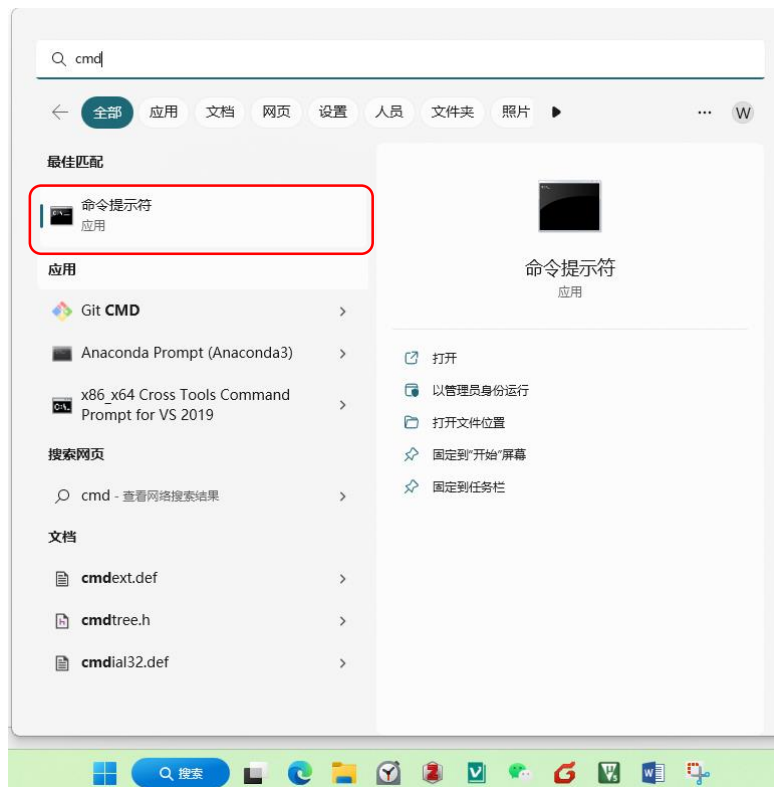
7、点击编辑按钮，弹出 Path 变量编辑对话框，如下图所示：



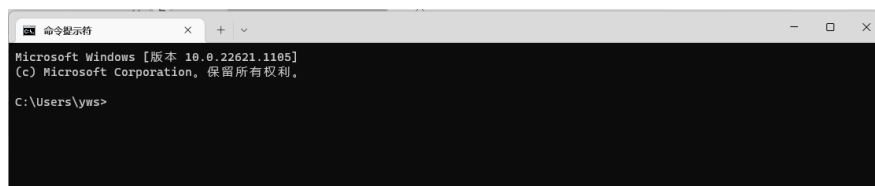
- 8、点击新建按钮，并将 MinGw-w64 的路径复制到新建栏。本文路径为：C:\Program Files\mingw-w64\x86\_64-8.1.0-release-posix-seh-rt\_v6-rev0\mingw64\bin。如下图所示：



9、打开命名提示符，如下图所示。

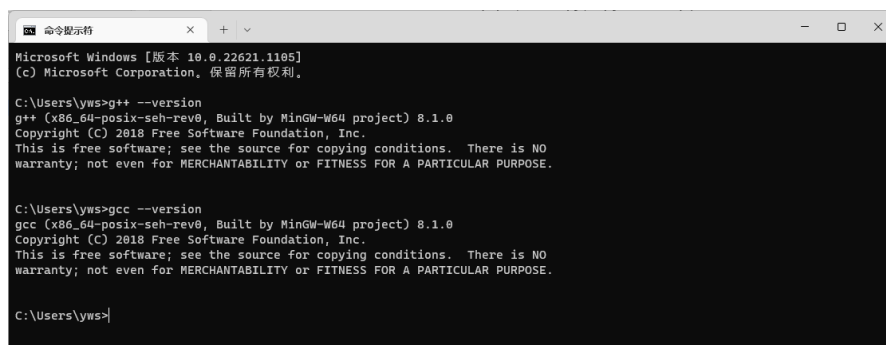


10、 点击命令提示符，得到如下窗口：



11、 在命令提示符窗口分别输入以下命令，得到如下输出，就表示基于 MinGW-w64 的 GNU 编程环境已经搭建成功。

命名： g++ --version  
gcc --version



### 三、编写第一个程序

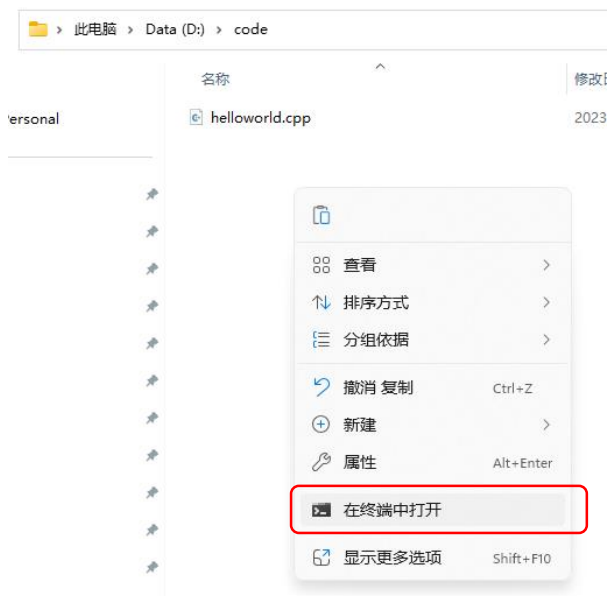
1) 在 D 盘创建文件夹: D:\code

- 2) 在 D:\code 文件夹内创建一个 C++ 源程序文件：helloworld.cpp。文件内容如下：

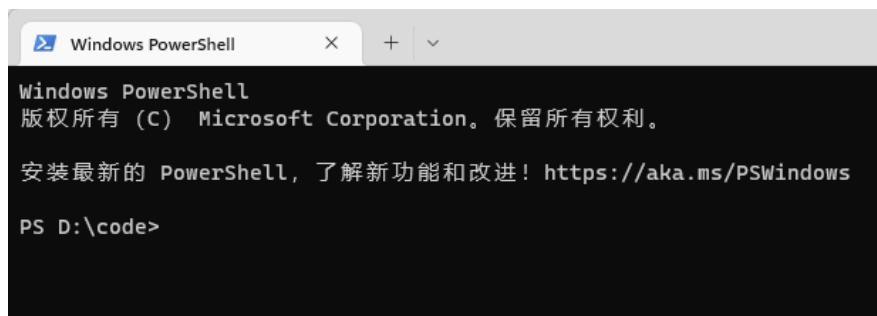
```
#include <iostream>
using namespace std;

int main()
{
    cout<<"Hello World!"<<endl;
    return 0;
}
```

- 3) 在 D:\code 目录下右击鼠标，选择在终端中打开。如下图所示：



- 4) 跳出命令行提示符窗口，此时窗口目录是 D:\code，如下图所示：



- 5) 在窗口里输入：dir 命令，可以显示在该目录下的文件夹和文件。如下图所示。

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

PS D:\code> dir

    目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----             2023/2/5      16:18           105 helloworld.cpp

PS D:\code>
```

- 6) 在窗口里输入: `g++ -o helloworld.exe .\helloworld.cpp` 并按回车, 完成编译, 创建 `helloworld.exe` 可执行程序。如下图所示。

```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

PS D:\code> dir

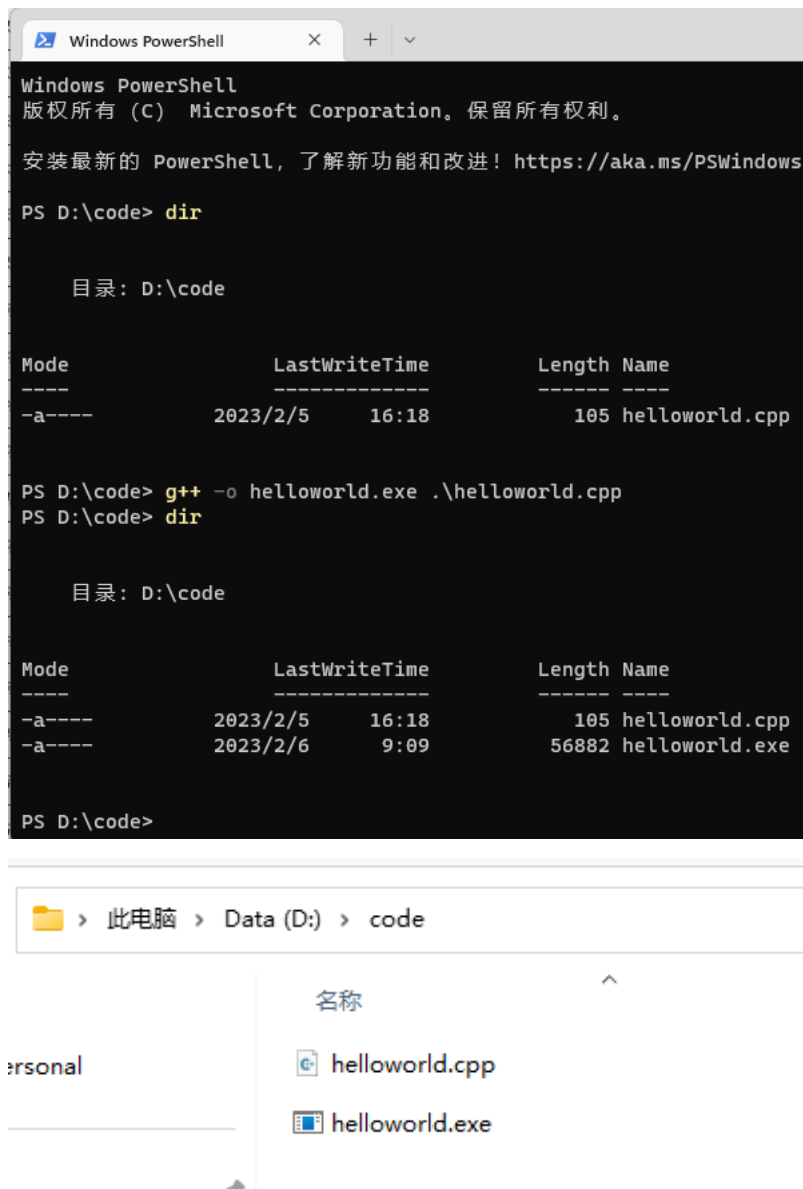
    目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----             2023/2/5      16:18           105 helloworld.cpp

PS D:\code> g++ -o helloworld.exe .\helloworld.cpp
PS D:\code>
```

此时在 `D:\code` 目录下生成了一个 `helloworld.exe` 文件。如下图所示。





- 7) 在命令行窗口输入 `.\helloworld.exe` 并按回车，则运行 `helloworld.exe` 程序。如下图所示。

```
Windows PowerShell X + v

安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

PS D:\code> dir

    目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a-----         2023/2/5          16:18         105 helloworld.cpp

PS D:\code> g++ -o helloworld.exe .\helloworld.cpp
PS D:\code> dir

    目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a-----         2023/2/5          16:18         105 helloworld.cpp
-a-----         2023/2/6           9:09       56882 helloworld.exe

PS D:\code> .\helloworld.exe
Hello World!
PS D:\code>
```

## 四、调试程序

### 1、语法错误

MinGW-w64 命令行形式能输出对应行的语法错误信息。在 D:\code 创建 C++ 源程序文件 twoNumMulti.cpp。代码如下:

```
#include <iostream>
using namespace std;

int main()
{
    int x,y;
```

```

cout<<"请输入两个整数:\n"
cin>>x>>y;
cout<<"两个整数的乘积为:"<<x*y<<endl;

return 0;
}

```

- 1) 在命令行提示符窗口编译 twoNumMulti.cpp 文件，生成 twoNumMulit.exe，如下图所示。

```

PS D:\code> dir

目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----            2023/2/5         16:18          105 helloworld.cpp
-a----            2023/2/6          9:09        56882 helloworld.exe
-a----            2023/2/6          9:52          170 twoNumMulti.cpp

PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
.\twoNumMulti.cpp: In function 'int main()':
.\twoNumMulti.cpp:8:27: error: expected ';' before 'cin'
    cout<<"请输入两个整数:\n"
                          ^
                          ;
    cin>>x>>y;
    ~~~~~
PS D:\code>

```

表示在语句（ cout<<"请输入两个整数:\n"）后面少了一个分号(;)。

- 2) 在该语句后面添加分号后，即可完成编译，生成 twoNumMulti.exe 可执行程序。如下图所示：

```

PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
.\twoNumMulti.cpp: In function 'int main()':
.\twoNumMulti.cpp:8:27: error: expected ';' before 'cin'
    cout<<"请输入两个整数:\n"
                          ^
                          ;
    cin>>x>>y;
    ~~~~~
PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
PS D:\code> dir

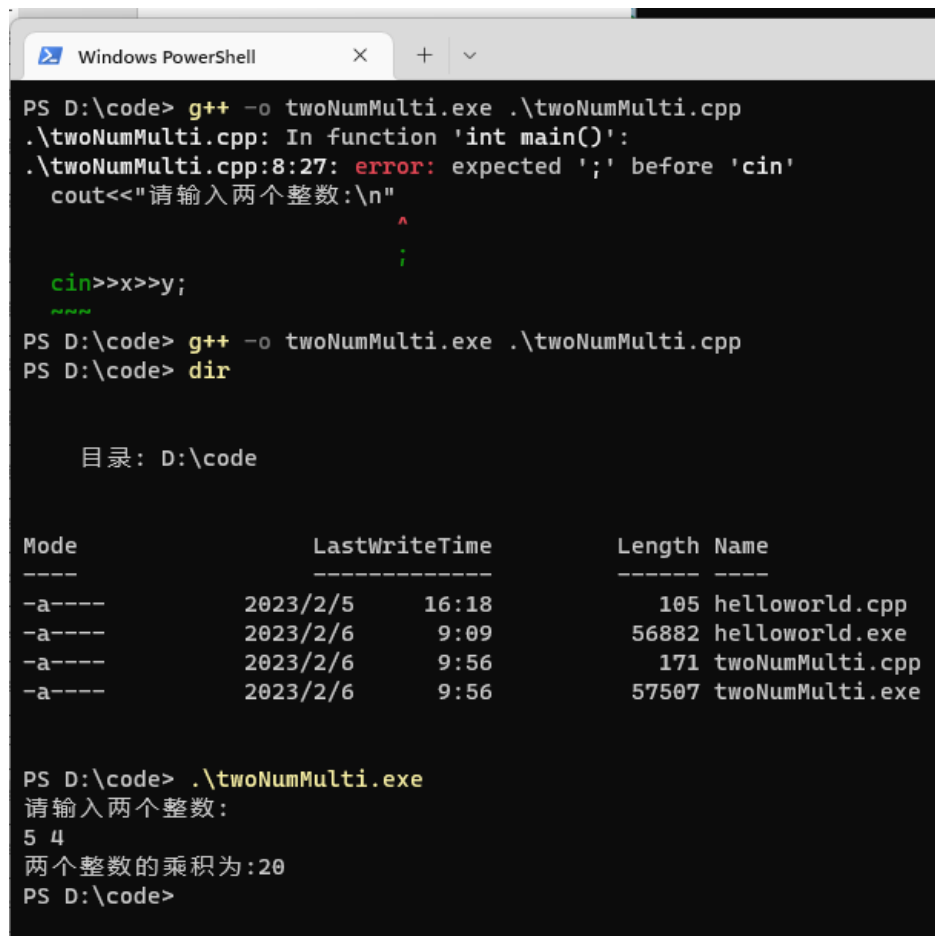
目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----            2023/2/5         16:18          105 helloworld.cpp
-a----            2023/2/6          9:09        56882 helloworld.exe
-a----            2023/2/6          9:56          171 twoNumMulti.cpp
-a----            2023/2/6          9:56        57507 twoNumMulti.exe

PS D:\code>

```

3) 运行 twoNumMulti.exe 程序。如下图所示：



```
PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
.\twoNumMulti.cpp: In function 'int main()':
.\twoNumMulti.cpp:8:27: error: expected ';' before 'cin'
    cout<<"请输入两个整数:\n"
                          ^
                          ;
    cin>>x>>y;
    ~~~~~
PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
PS D:\code> dir

    目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----            2023/2/5         16:18          105 helloworld.cpp
-a----            2023/2/6          9:09        56882 helloworld.exe
-a----            2023/2/6          9:56          171 twoNumMulti.cpp
-a----            2023/2/6          9:56        57507 twoNumMulti.exe

PS D:\code> .\twoNumMulti.exe
请输入两个整数:
5 4
两个整数的乘积为:20
PS D:\code>
```

## 2、逻辑错误

对前面的 twoNumMulti.cpp 程序做如下修改：

```
#include <iostream>
using namespace std;

int main()
{
    int x,y;
    cout<<"请输入两个整数:\n";
    cin>>x;
    cout<<"两个整数的乘积为:"<<x*y<<endl;

    return 0;
}
```

1) 编译程序 twoNumMulti.exe 并运行该程序。如下图所示。

```
Windows PowerShell
PS D:\code> dir

目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----          2023/2/5         16:18          105 helloworld.cpp
-a----          2023/2/6          9:09        56882 helloworld.exe
-a----          2023/2/6         10:01          168 twoNumMulti.cpp

PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
PS D:\code> dir

目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----          2023/2/5         16:18          105 helloworld.cpp
-a----          2023/2/6          9:09        56882 helloworld.exe
-a----          2023/2/6         10:01          168 twoNumMulti.cpp
-a----          2023/2/6         10:02        57507 twoNumMulti.exe

PS D:\code> .\twoNumMulti.exe
请输入两个整数:
5 4
两个整数的乘积为:0
PS D:\code>
```

2) 结果显然不正确。首先考虑  $x, y$  的值是否正确? 为此, 修改 twoNumNulit.cpp 文件如下。

```
#include <iostream>
using namespace std;

int main()
{
    int x,y;
    cout<<"请输入两个整数:\n";
    cin>>x;
    cout<<"x="<<x<<"\ty="<<y<<endl;
    cout<<"两个整数的乘积为:"<<x*y<<endl;

    return 0;
}
```

增加的语句是为了查看在计算  $x*y$  之前  $x$  和  $y$  的值。

3) 再次重新生成 twoNumMulti.exe, 并运行该程序。如下图所示。

```
Windows PowerShell
PS D:\code> dir

目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----            2023/2/5         16:18          105 helloworld.cpp
-a----            2023/2/6          9:09        56882 helloworld.exe
-a----            2023/2/6         10:04          202 twoNumMulti.cpp
-a----            2023/2/6         10:02       57507 twoNumMulti.exe

PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
PS D:\code> .\twoNumMulti.exe
请输入两个整数:
5 4
x=5      y=0
两个整数的乘积为:0
PS D:\code>
```

4) 从上面结果看出, y 并没有接收输入的整数 4, 说明对 y 输出数据有错误, 检查源程序文件, 发现没有输入 y 的值。修改此语句。

```
cin>>x;    =>    cin>>x>>y;
```

5) 再次重新生成 twoNumMulti.exe, 并运行该程序。如下图所示。

```
Windows PowerShell
PS D:\code> dir

目录: D:\code

Mode                LastWriteTime         Length Name
----                -
-a----            2023/2/5         16:18          105 helloworld.cpp
-a----            2023/2/6          9:09        56882 helloworld.exe
-a----            2023/2/6         10:10          205 twoNumMulti.cpp
-a----            2023/2/6         10:10       58019 twoNumMulti.exe

PS D:\code> g++ -o twoNumMulti.exe .\twoNumMulti.cpp
PS D:\code> .\twoNumMulti.exe
请输入两个整数:
5 4
x=5      y=4
两个整数的乘积为:20
PS D:\code>
```

6) 程序正确。OK。

## 五、Visual Studio Code 编程环境配置

Visual Studio Code 编程环境配置是在前面 C/C++ 编程环境搭建完成的基础上完成 C/C++ 编程环境配置。如果第二步(二、编程环境搭建)没有完成, 请先完成第二步, 然后再配置 VSCode 的编程环境。

## 1、按照 VSCode 和相应扩展插件

- 1) 安装 Visual Studio Code(VSCode)。
- 2) 安装插件：点击扩展显示已安装的插件。如下图所示。



为 VSCode 安装上面的扩展插件。

## 2、运行一个程序

- 1) 在 VSCode 中打开一个目录，本文为 D:\code2 目录，并创建一个文件 helloworld.cpp。如下图所示：



- 2) 打开 helloworld.cpp 文件，并输入如下代码：

```
#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main()
```

```

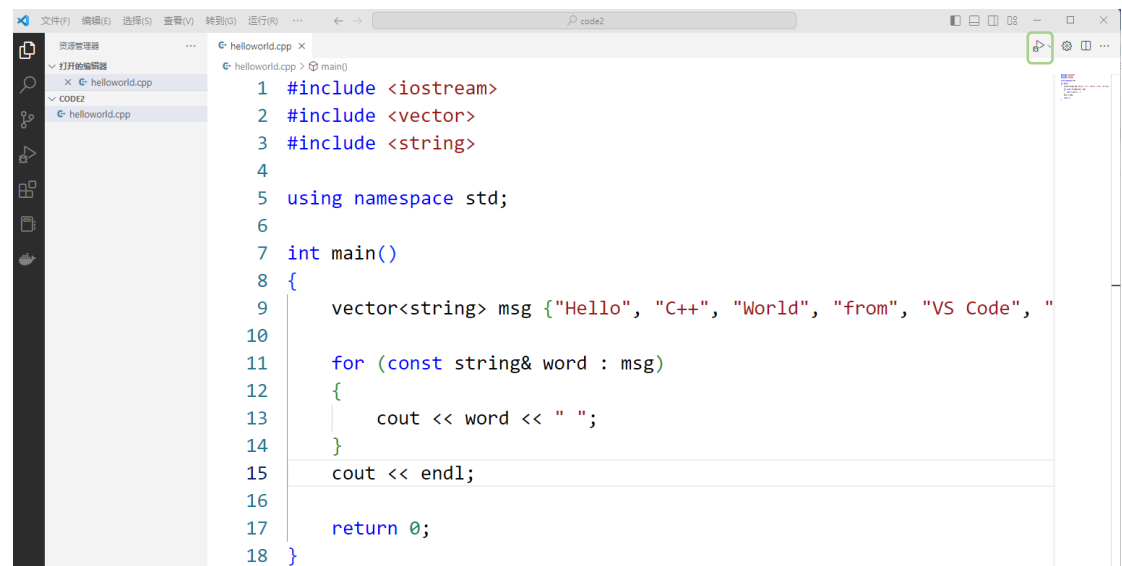
{
    vector<string> msg { "Hello", "C++", "World", "from", "VS Code", "and the C++ extension!" };

    for (const string& word : msg)
    {
        cout << word << " ";
    }
    cout << endl;

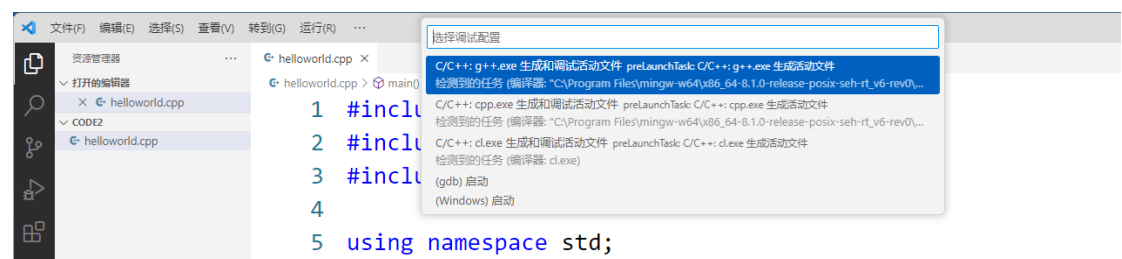
    return 0;
}

```

3) 点击运行按键，如下图所示。



4) 从计算机系统里检测到的编译器列表中选择 C/C++: g++.exe 生成和调试活动文件。如下图所示。



这里只有在运行目录下的第一个 C++ 程序文件时需要选择编译器，然后 VSCode 会生成默认的 `tasks.json`，可以在你的项目目录下的 `.vscode` 文件夹里找到 `tasks.json`。这个文件保存了 VSCode 的编译配置参数。

5) 产生的 `tasks.json` 文件与下面所示的 JSON 内容类似。

```

{
    "tasks": [

```



```

{
  "type": "cppbuild",
  "label": "C/C++: g++.exe 生成活动文件",
  "command": "C:\\Program Files\\mingw-w64\\x86_64-8.1.0-release-posix-seh-rt_v6-
rev0\\mingw64\\bin\\g++.exe",
  "args": [
    "-fdiagnostics-color=always",
    "-g",
    "${file}",
    "-o",
    "${fileDirname}\\${fileBasenameNoExtension}.exe"
  ],
  "options": {
    "cwd": "${fileDirname}"
  },
  "problemMatcher": [
    "$gcc"
  ],
  "group": {
    "kind": "build",
    "isDefault": true
  },
  "detail": "调试器生成的任务。"
}
],
"version": "2.0.0"
}

```

其中：

- **command 参数**：规定了将要运行的程序，上面文件中表示要运行 g++.exe。这里需要 g++.exe 的完整路径。
- **args 参数**：规定了要传递给 g++ 的命令行参数，这些参数必须按照编译器要求的顺序进行传递。
  - **\${file}**：告诉 g++ 要编译的文件
  - **\${fileDirname}**：指定了 g++ 当前激活文件的路径以及生成可执行程序保存的路径
  - **\${fileBasenameNoExtension}.exe**：指定了可执行程序文件的文件名与激活文件同名，只是后缀名不同。例如上面文件为 helloworld.exe
- **group 参数**：如果设置了 isDefault 为 true，则将对这个工程项目下的所有文件都采用这个默认配置进行编译。
- 如果要编译多个文件，可以将参数 "\${file}" 修改为 "\${workspaceFolder}/\*.cpp"。
- 也可以将 "\${fileDirname}\\\${fileBasenameNoExtension}.exe" 修改为指定路径下的可执行程序文件名，例如： "\${workspaceFolder}\\myProgram.exe"。

6) 运行程序结果如下图所示：

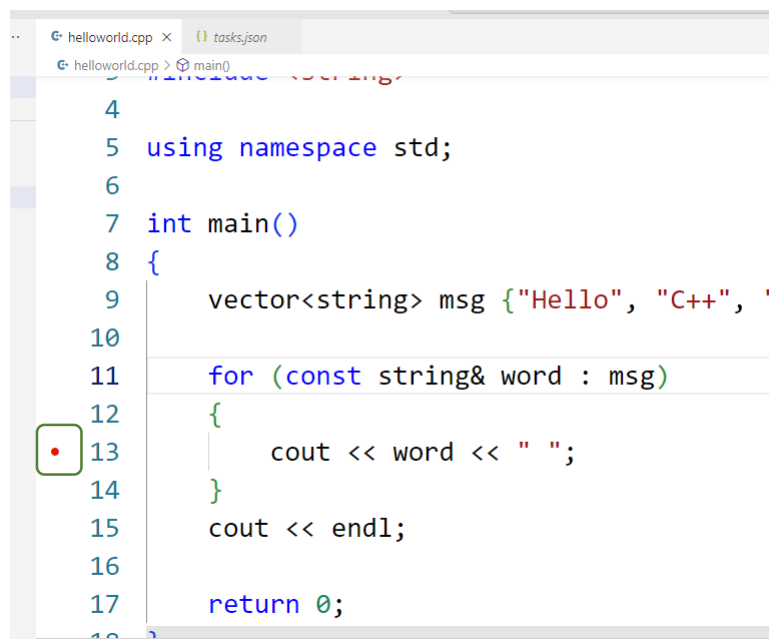
```
D:\code2> cmd /C "c:\Users\yws\.vscode\extensions\ms-vscode.cpptools-1.
rs2r0.txj --stderr=Microsoft-MIEngine-Error-d2thwpy1.hsx --pid=Microsof
reter=mi "
```

```
D:\code2> cmd /C "c:\Users\yws\.vscode\extensions\ms-vscode.cpptools-1.
0hymt.fwx --stderr=Microsoft-MIEngine-Error-lhdkjos3.kaa --pid=Microsof
reter=mi "
```

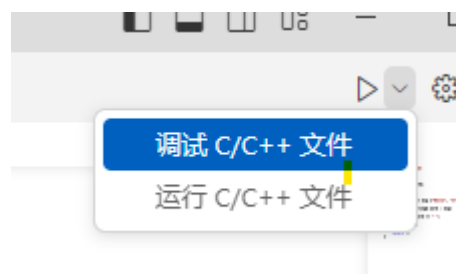
Hello C++ World from VS Code and the C++ extension!

### 3、调试程序

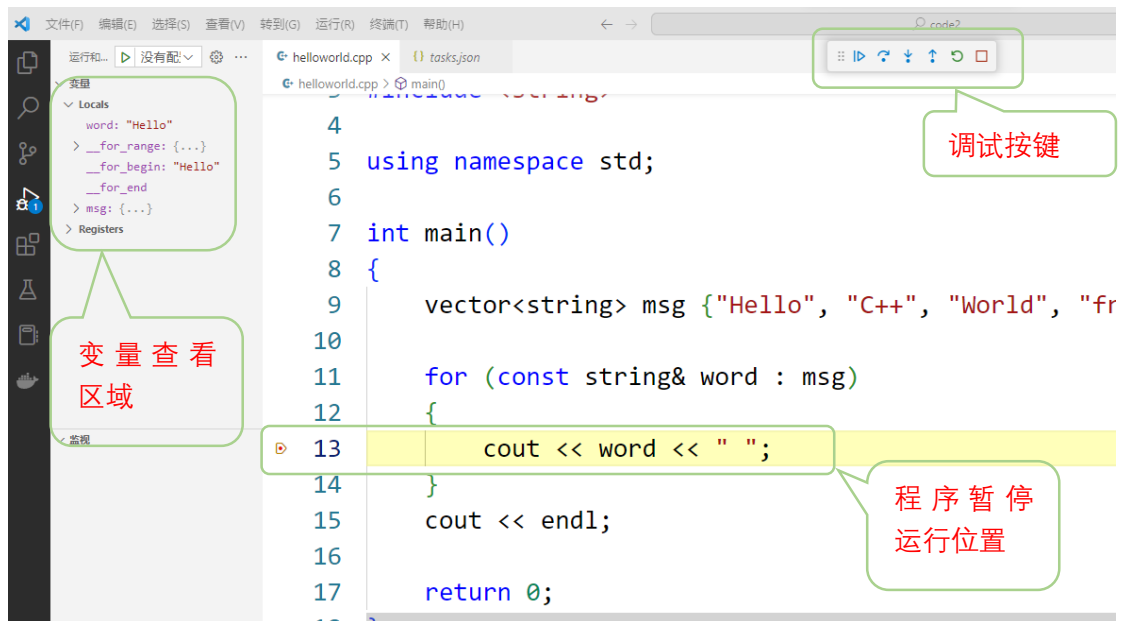
- 1) 再次打开 helloworld.cpp
- 2) 设置断点：在 helloworld.cpp 文件对应行左边窗口栏的双击鼠标，可以设置断点。如下图所示。



- 3) 在右侧运行按钮的下拉框中选择调试 C/C++ 文件即可启动调试。如下图所示。



- 4) 程序运行到断点位置，可以查看当前的变量值。如下图所示。



调试按键的具体功能可以将鼠标放到按键位置查看。

VSCode 的编程环境配置详细内容可查看如下网址：

<https://code.visualstudio.com/docs/cpp/config-mingw>